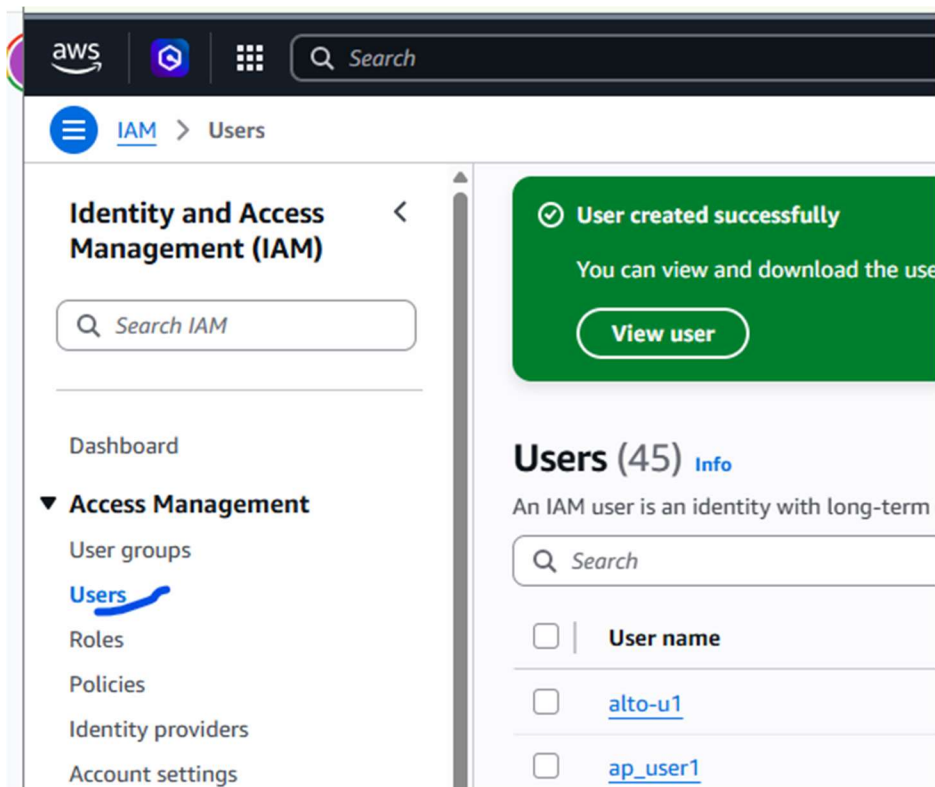


Create IAM user

Steps

1. Create iam group/user with ec2 full access
2. Create access key and secret

1. Create iam group/user with ec2 full access



Click on user and then on the right side , click on “Create user”

Step 1: Specify user details

Step 2: Set permissions

Step 3: Review and create

Specify user details

User details

User name:

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and +, =, @, _ (hyphen)

☐ Provide user access to the AWS Management Console - optional

In addition to console access, users with `SignInLocalDevelopmentAccess` permissions can use the same console credentials for programmatic access without the need for access keys.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user.

[Learn more](#)

Cancel Next

Click Next.

Step 1: Specify user details

Step 2: Set permissions

Step 3: Review and create

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Permissions options

☒ Add user to group

Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

☐ Copy permissions

Copy all group memberships, attached managed policies, and inline policies from an existing user.

☐ Attach policies directly

Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

User groups (5)

<input type="checkbox"/>	Group name	Users	Attached policies	Created
<input type="checkbox"/>	Admin-Group-A2	1	AdministratorAccess	2025-04-04 (10 months ago)
<input type="checkbox"/>	b06-g1	1	AmazonEC2ReadOnlyAccess	2025-08-18 (6 months ago)
<input type="checkbox"/>	g1	0	vpc-full-access-may25	2025-05-11 (9 months ago)
<input type="checkbox"/>	group1	1	AmazonEC2FullAccess	2025-09-08 (5 months ago)
<input type="checkbox"/>	specnt-tf	31	-	2025-01-31 (1 year ago)

[Create group](#)

Set permissions boundary - optional

Cancel Previous Next

Click on “Attach policies directly”

And in the permission search for “ec2full”

Step 1

Specify user details

Step 2

Set permissions

Step 3

Review and create

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

☐ Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

☐ Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.

☒ Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1590)

Choose one or more policies to attach to your new user.

Filter by Type

All types

1 match

<input type="checkbox"/>	Policy name	Type	Attached entities
<input type="checkbox"/>	AmazonEC2FullAccess	AWS managed	6

► Set permissions boundary - optional

Cancel

Previous

Next

Put a tick mark and click on next.

Step 1

Specify user details

Step 2

Set permissions

Step 3

Review and create

Review and create

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

User details

User name	Console password type	Require password reset
tf-user01	None	No

Permissions summary

Name	Type	Used as
AmazonEC2FullAccess	AWS managed	Permissions policy

Tags - optional

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel

Previous

Create user

Click on Create user.

2. Create access key and secret

Click on the user.

The screenshot shows the AWS IAM console interface. On the left is a navigation sidebar with 'Identity and Access Management (IAM)' selected. The main content area displays the details for user 'tf-user01'. The 'Summary' tab is active, showing the user's ARN, console access status (Disabled), and creation date. Below this, the 'Permissions' tab is selected, showing a list of permissions policies. A table lists the policy 'AmazonEC2FullAccess' with a type of 'AWS managed'.

Policy name	Type	Attached to
AmazonEC2FullAccess	AWS managed	Direct

Click on “Security credentials”

This screenshot shows the 'Security credentials' tab for the user 'tf-user01'. It contains three main sections: 'Console sign-in', 'Multi-factor authentication (MFA)', and 'Access keys'. The 'Console sign-in' section shows a disabled console password and an 'Enable console access' button. The 'MFA' section indicates that no MFA devices are currently assigned, with an 'Assign MFA device' button. The 'Access keys' section shows that no access keys are present, with a 'Create access key' button. A blue line points to this button in the original image.

Click on “Creaet access key”

Step 1

Access key best practices & alternatives

Step 2 - optional

Set description tag

Step 3

Retrieve access keys

Access key best practices & alternatives [Info](#)

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

Use case

- ☐ **Command Line Interface (CLI)**
You plan to use this access key to enable the AWS CLI to access your AWS account.
- ☐ **Local code**
You plan to use this access key to enable application code in a local development environment to access your AWS account.
- ☐ **Application running on an AWS compute service**
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.
- ☐ **Third-party service**
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.
- ☐ **Application running outside AWS**
You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.
- ☐ **Other**
Your use case is not listed here.

[Cancel](#)[Next](#)

Select the first option “command line interface”

Click on **next**

Create access key

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

tag

keys

Use case

- ☒ **Command Line Interface (CLI)**
You plan to use this access key to enable the AWS CLI to access your AWS account.
- ☐ **Local code**
You plan to use this access key to enable application code in a local development environment to access your AWS account.
- ☐ **Application running on an AWS compute service**
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.
- ☐ **Third-party service**
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.
- ☐ **Application running outside AWS**
You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.
- ☐ **Other**
Your use case is not listed here.

**Alternatives recommended**

- Use AWS CLI V2 and the `aws login` command to use your existing console credentials in the CLI. [Learn more](#)
- Use AWS CloudShell, a browser-based CLI, to run commands. [Learn more](#)

Confirmation☒ I understand the above recommendation and want to proceed to create an access key.[Cancel](#)[Next](#)

AM > Users > tf-user01 > Create access key

Step 1

Access key best practices & alternatives

Step 2 - optional

Set description tag

Step 3

Retrieve access keys

Set description tag - optional [Info](#)

The description for this access key will be attached to this user as a tag and shown alongside the access key.

Description tag value
Describe the purpose of this access key and where it will be used. A good description will help you rotate this access key confidently later.

ec2fullaccess

Maximum 256 characters. Allowed characters are letters, numbers, spaces representable in UTF-8, and: _ . : / = + - @

[Cancel](#) [Previous](#) [Create access key](#)

Click on “create access key”.

&

Retrieve access keys [Info](#)

Access key
If you lose or forget your secret access key, [create a new access key and make the old key inactive.](#)

Access key

AKIAWOXUG33LXBXITMC

Secret access key

***** [Show](#)

✔ Secret access key Copied

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [best practices for managing AWS access keys](#).

[Download .csv file](#) [Done](#)

Copy the secret and save it , as it would NOT be available again

Or download the .csv file